



Wall

壁 (Wall)

Jian-Jia is building a wall by stacking bricks of the same size together. This wall consists of n columns of bricks, which are numbered 0 to $n-1$ from left to right. The columns may have different heights. The height of a column is the number of bricks in it.

ジェンジア (Jian-Jia) は同じ大きさのレンガを積み上げて壁を作っている。壁は n 列からなり、列には左から右へ順に 0 から $n-1$ までの番号が付けられている。ある列の高さとはその列にあるレンガの個数であり、それぞれの列の高さはすべて同じである必要はない。

Jian-Jia builds the wall as follows. Initially there are no bricks in any column. Then, Jian-Jia goes through k phases of *adding* or *removing* bricks. The building process completes when all k phases are finished. In each phase Jian-Jia is given a range of consecutive brick columns and a height h , and he does the following procedure:

ジェンジアは次のようにして壁を作る。はじめはどの列にもレンガは無い。そこからジェンジアはレンガを *加える* か *取り除く* といった操作を k 回行う。 k 回の操作がすべて終わると壁の建築は完了する。それぞれの操作においてジェンジアは列のある連続した範囲と高さ h を指定され、以下のような手続きを行う。

- In an *adding* phase, Jian-Jia adds bricks to those columns in the given range that have less than h bricks, so that they have exactly h bricks. He does nothing on the columns having h or more bricks.
- *加える* 操作では、ジェンジアは指定された列のうち、レンガの個数が h に満たないような列に対してレンガの個数がちょうど h となるようにレンガを加える。レンガの個数が h 以上であるような列に対しては何も行わない。
- In a *removing* phase, Jian-Jia removes bricks from those columns in the given range that have more than h bricks, so that they have exactly h bricks. He does nothing on the columns having h bricks or less.
- *取り除く* 操作では、ジェンジアは指定された列のうち、レンガの個数が h よりも大きいような列に対してレンガの個数がちょうど h となるようにレンガを取り除く。レンガの個数が h 以下であるような列に対しては何も行わない。

Your task is to determine the final shape of the wall.

壁の最終的な形を決定するプログラムを作成せよ。

Example

例 (Example)

We assume that there are 10 brick columns and 6 wall building phases. All ranges in the following table are inclusive. Diagrams of the wall after each phase are shown below.

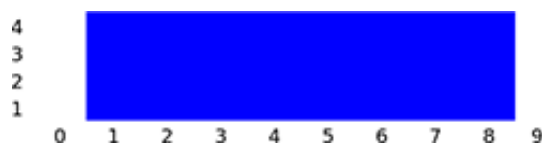
壁は 10 個の列からなり，建築は 6 個の操作によって行われるとする．以下の表に示される範囲はすべて両端を含む．下にある図はそれぞれの操作の後の壁の様子を表している．

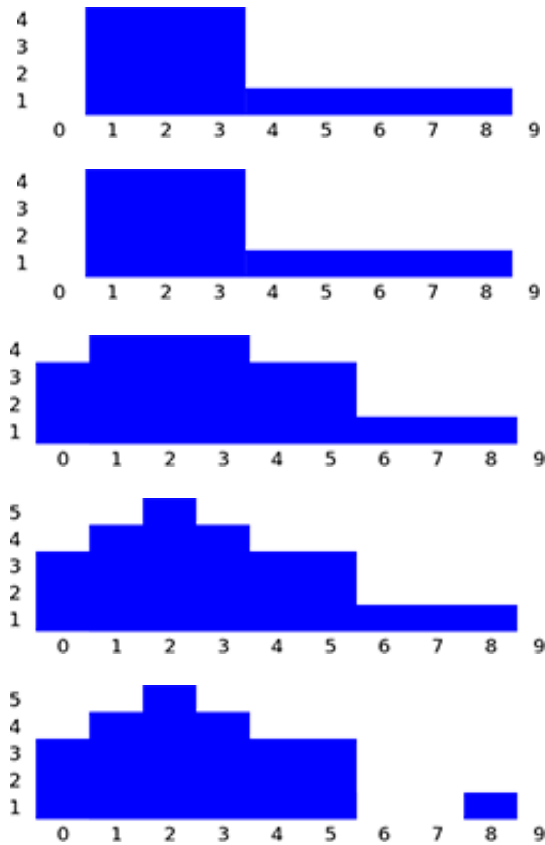
phase	type	range	height
0	add	columns 1 to 8	4
1	remove	columns 4 to 9	1
2	remove	columns 3 to 6	5
3	add	columns 0 to 5	3
4	add	column 2	5
5	remove	columns 6 to 7	0

操作	種類	範囲	高さ
0	加える	列 1 から列 8	4
1	取り除く	列 4 から列 9	1
2	取り除く	列 3 から列 6	5
3	加える	列 0 から列 5	3
4	加える	列 2	5
5	取り除く	列 6 から列 7	0

Since all columns are initially empty, after phase 0 each of the columns 1 to 8 will have 4 bricks. Columns 0 and 9 remain empty. In phase 1, the bricks are removed from columns 4 to 8 until each of them has 1 brick, and column 9 remains empty. Columns 0 to 3, which are out of the given range, remain unchanged. Phase 2 makes no change since columns 3 to 6 do not have more than 5 bricks. After phase 3 the numbers of bricks in columns 0, 4, and 5 increase to 3. There are 5 bricks in column 2 after phase 4. Phase 5 removes all bricks from columns 6 and 7.

はじめはどの列にもレンガがないため，操作 0 の後では列 1 から列 8 にはそれぞれ 4 個のレンガが積まれており，列 0 と列 9 にはレンガがないままである．操作 1 では，列 4 から列 8 にあるレンガがそれぞれの列にあるレンガが 1 個になるまで取り除かれ，列 9 はレンガがないままである．また，列 0 から列 3 は指定された範囲の外にあるためレンガの個数は変わらない．操作 2 では，列 3 から列 6 はどの列もレンガの個数が 5 より大きくないため何も行われない．操作 3 が終わった後では列 0, 4, 5 のレンガが 3 個に増やされている．操作 4 が終わった後では列 2 に 5 個のレンガがある．操作 5 では列 6 と列 7 にあるレンガがすべて取り除かれる．





Task

課題 (Task)

Given the description of the k phases, please calculate the number of bricks in each column after all phases are finished. You need to implement the function `buildWall`.

k 個の操作の情報が与えられたとき、すべての操作が終わった後に壁のそれぞれの列にレンガがいくつあるかを求めよ。関数 `buildWall` を実装せよ。

- `buildWall(n, k, op, left, right, height, finalHeight)`
 - `n`: the number of columns of the wall.
 - `k`: the number of phases.
 - `op`: array of length k ; `op[i]` is the type of phase i : 1 for an adding phase and 2 for a removing phase, for $0 \leq i \leq k-1$.
 - `left` and `right`: arrays of length k ; the range of columns in phase i starts with column `left[i]` and ends with column `right[i]` (including both endpoints `left[i]` and `right[i]`), for $0 \leq i \leq k-1$. You will always have `left[i] ≤ right[i]`.
 - `height`: array of length k ; `height[i]` is the height parameter of phase i , for $0 \leq i \leq k-1$.
 - `finalHeight`: array of length n ; you should return your results by placing the final number of bricks in column i into

$\text{finalHeight}[i]$, for $0 \leq i \leq n - 1$.

- $\text{buildWall}(n, k, \text{op}, \text{left}, \text{right}, \text{height}, \text{finalHeight})$
 - n : 壁の列の個数
 - k : 操作の回数
 - op : 長さが k の配列であり, $0 \leq i \leq k - 1$ に対し $\text{op}[i]$ は操作 i の種類を表す. 1 は加える操作を, 2 は取り除く操作を表す.
 - left および right : どちらも長さが k の配列であり, $0 \leq i \leq k - 1$ に対し, 操作 i で指定される列の範囲が列 $\text{left}[i]$ から列 $\text{right}[i]$ であることを表す (両端の $\text{left}[i]$ と $\text{right}[i]$ も含む). $\text{left}[i] \leq \text{right}[i]$ が常に成り立つ.
 - height : 長さが k の配列であり, $0 \leq i \leq k - 1$ に対し $\text{height}[i]$ は操作 i で指定される高さを表す.
 - finalHeight : 長さが n の配列であり, $0 \leq i \leq n - 1$ に対し, 最終的に列 i にあるレンガの個数を $\text{finalHeight}[i]$ に設定することで結果を返さねばならない.

Subtasks

小課題 (Subtasks)

For all subtasks the height parameters of all phases are nonnegative integers less or equal to **100,000**.

すべての小課題において操作の際に与えられる高さは **100,000** 以下の非負整数である.

subtask	points	n	k	note
1	8	$1 \leq n \leq 10,000$	$1 \leq k \leq 5,000$	no additional limits
2	24	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	all adding phases are before all removing phases
3	29	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	no additional limits
4	39	$1 \leq n \leq 2,000,000$	$1 \leq k \leq 500,000$	no additional limits

小課題	得点	n	k	追加の制約
1	8	$1 \leq n \leq 10,000$	$1 \leq k \leq 5,000$	追加の制約はない

2 小 題	29 得 点	$1 \leq n \leq 100,000$ n	$1 \leq k \leq 500,000$ k	サブプログラムの操作はこの取り除く操作による追加の制約
3		$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	追加の制約はない
4	39	$1 \leq n \leq 2,000,000$	$1 \leq k \leq 500,000$	追加の制約はない

Implementation details

実装の詳細 (Implementation details)

You have to submit exactly one file, called `wall.c`, `wall.cpp` or `wall.pas`. This file implements the subprogram described above using the following signatures. You also need to include a header file `wall.h` for C/C++ program.

1 つのファイルを提出せよ。提出するファイルの名前は `wall.c`, `wall.cpp`, `wall.pas` のいずれかである。このファイルには課題で指定されたサブプログラムを以下のシグネチャを用いて実装すること。C または C++ のプログラムにおいては、`wall.h` をインクルードする必要がある。

C/C++ program

C/C++ プログラム (C/C++ program)

```
void buildWall(int n, int k, int op[], int left[], int right[],
int height[], int finalHeight[]);
```

Pascal program

Pascal プログラム (Pascal program)

```
procedure buildWall(n, k : longint; op, left, right, height :
array of longint; var finalHeight : array of longint);
```

Sample grader

採点プログラムのサンプル (Sample grader)

The sample grader reads the input in the following format:
採点プログラムのサンプルは以下の形式の入力を読み込む:

- line 1: n, k .
- line $2 + i$ ($0 \leq i \leq k - 1$): $op[i], left[i], right[i], height[i]$.
- 1 行目: n, k .
- $2 + i$ 行目 ($0 \leq i \leq k - 1$): $op[i], left[i], right[i], height[i]$.